DevOps for Information Systems: A Review of the State-of-the-Art

Asif Q. Gill, Abhishek Loumish, Isha Riyat and, Sung Y. Han, University of Technology Sydney (UTS), *Australia.*

Abstract— There is growing interest among organizations in contemporary DevOps practices and technology. Despite this growing interest, organizations are unsure how to effectively establish a DevOps capability for the continuous delivery of information systems including software and hardware that support data-intensive applications. Thus, the purpose of this paper is to compile and analyze the state-of-the-art in DevOps. Applying the well-known systematic literature review (SLR) approach and customized search criteria, we initially identified 3,790 papers and of which final 32 relevant papers were selected and reviewed in this qualitative research study. The results are organized using the well-known ISO/IEC 24744 metamodel elements: people (roles), process, technology and artefacts. We found 11 major roles, 6 processes, 23 technolgies, 5 artefacts, and 7 challenges (including 6 corresponding solutions). DevOps engineer is being the newly identified role. Continuous delivery pipeline and continuous improvement are the most highlighted major DevOps processes. Build system technology is being the key focus of the DevOps. Finally, major challenges are around people and culture, and the misunderstanding of the DevOps. Potential research areas for further research are: DevOps analytics, artefacts and tool-chain integration. The findings of this research will serve as a resource for both practitioners and researchers who have interest in DevOps. This paper is limited to the reviewed studies and selected well-known four scholarly and two industry databases.

Index Terms— DevOps, Continuous Delivery, Continuous Deployment, Challenges, Information Systems, Systematic Literature Review

1 INTRODUCTION

Agile approaches focus on the continuous delivery of working information systems (IS) in small iterations (Agile Manifesto 2001). In a typical IS delivery environment, a separate operations team is required to deploy the working software and hardware in production for data-intensive applications (e.g. IoT, Mobile, BigData and Datawarehouse Applications). The emerging DevOps movement, in the context of agile software delivery, includes operations in development to make available the working software and hardware to customers as quickly as possible without any undesired delays (Bass et al. 2015). For the continuous and fast deployment of software and hardware, it is important to do necessary deployment automation and bridge two silos by including the operations in development (Hüttermann 2012; Wettinger 2013).

The continuous delivery can be governed by the effective and efficient coordination between the development (e.g. software) and operations (e.g. hardware) teams (Peuraniemi 2014). However, most of the organizations have two separate capabilities in the form of development and operations, which is perceived as a challenge for fast and frequent releases (Wettinger et al. 2014). DevOps is gaining momentum and is considered as a prominent area of research and practice. However, this term is often referred to as 'ambiguous' due to its varied perceptions and interpretations (Peuraniemi 2014; Liu et. al. 2014). DevOps is considered as an extension of agile methodologies from development to systems operation; a combination of development and operations; or it is a way to manage the end-to-end product lifecycle (Patwardhan 2012; Peuraniemi 2014). The integrated DevOps capability could be seen as an alternative to the traditional cultural, collaborative and management problems in the isolated development and operations capabilities (Rackspace 2014; Erich et al. 2014). However, the establishment of DevOps capability is not a simple straightforward task and involves the consideration of a number of elements (Claps el al. 2015). This draws our attention to the following main research question:

RQ: How to effectively establish a DevOps capability for the continuous delivery of IS including software and hardware?

In order to address this important research question, it is important to understand the DevOps capability elements and their details. Thus, a Systematic Literature Review (SLR) (Kitchenham & Charters 2007) was conducted and synthesized using the four core high level elements of people, process, technology and artefact (work product) – adopted from the well-known ISO/ IEC 24744 (2014) metamodel for development methodologies. The review results are organized around these four core elements and offer important consolidated taxonomy and knowledge-base, which would be helpful for establishing a DevOps capability. To the best of our knowledge, there are no such recent studies published in the public domain (at least, at the time that this study was first initiated) that offer such systematic review in the context of DevOps capability establishment. This systematic SLR approach to identifying, analysing and synthesizing the 'DevOps' literature will provide a consoli-

dated knowledge-base for researchers and practitioners. This paper also lays a foundation for further research in this important area.
The structure of this paper is organized as follows. Firstly, it provides the research background. Secondly, it discusses the SRL research method. Thirdly, it provides the research results. Finally, it discusses the results and conclusions.

2 BACKGROUND

In a typical IS delivery environment, development (Dev) and operations (Ops) have their own roles, processes and tools; and therefore, both Dev and Ops are setup and run as independent capabilities (Armenise, 2015). Wettinger et al. (2014) & Bayser et al. (2015) mention about the different or contrary goals and mindsets of the independent Dev and Ops teams. For instance, developers are keen on asserting changes to the operations, however operations want their environment to be stable. This kind of mindset or conflicting goals between these groups may lead to slow, manual and erroneous processes of IS delivery. To address these challenges, DevOps concept has been proposed, which suggests the adoption of an integrated DevOps approach for continuous delivery of IS (Peuraniemi 2014; Mueller et al. 2014). The principles of DevOps are highly based on automation, continuous delivery, short release cycles and agile development to enable delivering great value to the customer (Bayser et al. 2015; Neely & Stolt 2013). According to Peuraniemi (2014), DevOps facilitates early customer feedback loops, which may help to effectively address customer satisfaction concerns. Thus, it enables rapid and continuous feedback from the customer and identification of concise customer needs (Poppendieck & Cusumano 2012; Williams 2012).

DevOps offers alternative ways of working and focuses on simplicity, agility, measurable business values, and high-quality service delivery (Httermann 2012). It is a way to harvest trust and shared ownership in teams by breaking the undesired barriers, fostering innovation, and encouraging collaboration between concerned stakeholders (Farroha & Farroha 2014). Chen (2015b) also discusses the importance of integrated DevOps architecture for the continuous delivery of a portfolio of twenty-two applications in a case study organization. Farroha & Farroha (2014) discuss the important considerations of automation, organization culture and stakeholders' mindset for adopting new ways of working such as the DevOps. DevOps is a way to conduct IS development based on experiments and active engagement with stakeholders consisting of iterative feedback-driven build-measure-learn cycles (Bosch 2012; Fagerholm et al. 2014).

DevOps ways of working focus on enabling effective communication and collaboration (Gottesheim 2015). The efficient and effective communication and coordination between all stakeholders can be enabled through integrated DevOps, which is important for a successful IS delivery (Cois et al. 2014). It is important to have a flawless delivery pipeline, which can be optimized to the largest possible extent. It is also crucial to satisfy the continuous customer engagement and expectations of being responsive to the quick market changes, in any business environment with the time and the resource constraints (Stillwell et al. 2015). Virmani (2015) discusses the effective implementation of DevOps for productivity gains and value delivery to customers.

DevOps seems to offer considerable benefits, however, it also poses several challenges such as organizational, process, and technical challenges (Chen 2015a; Claps el al. 2015; Gottesheim 2015); for instance, organizational challenges may refer to organizational culture, enterprise data models, IT operating models, reward models, and risk allocation models. These challenges may hinder fostering measurable incremental changes to derive value from people, process and technology (McCarthy et al. 2015; Zhong et. al. 2012). Bellomo (2014) points out that one of the important challenge is to specify precise requirements of deployability both software hardware, which is an important pre-requisite for DevOps. Furthermore, Cois et al. (2014) highlighted that designing a practical software development process, supported by appropriate technology hardware, for DevOps is the real challenge for everyone. According to Hussaini, (2014), "leveraging the critical success factors to deliver the change must have shared objectives of Dev and Ops." Moreover, these objectives drive the process and people involved in the DevOps.

It is clear from the above mentioned recent studies that there is a significant interest in DevOps, and the adoption of DevOps is not a simple task. There is a need for clear understanding and guidelines to support the effective DevOps adoption for IS.

3 RESEARCH METHOD

This paper applied a SLR approach (Kitchenham & Charters 2007; Rowe 2014) to identify and synthesize the literature published in the context of DevOps adoption. According to Kitchenham et al. (2007), SLR is a systematic and structured approach to analyzing, selecting and synthesizing the literature admissible to the research question. Guidelines for citation and evaluation procedures (based on Dybå and Dingsøyr (2007)) have also been adopted in this paper to ensure the quality of the papers selected for this study. This study has been conducted in following distinct stages:

- Inclusion and exclusion criteria
- Selection of data sources, search strategies, and management of citation and inclusion decision
- Final study selection and quality assessment
- Review, data extraction and synthesis

3.1. Inclusion and exclusion criteria

The inclusion and exclusion criteria were guided by the following main questions.

RQ: How to effectively establish a DevOps capability for the continuous delivery of IS including software and hardware?

- **RQ1:** What are the key people (roles), processes, technology and artefacts (workproducts) for DevOps?
- **RQ2:** What are the key challenges in adopting DevOps?
- **RQ3***:* What are the possible solutions to the challenges in adopting DevOps?

These important questions guide our study. We adopted four core elements (as discussed earlier) for DevOps capability from well-known ISO/ IEC 24744 (2014) metamodel. These elements from the ISO/IEC theoretical metamodel are used to extract the DevOps people (roles), process, technology and artefact (workproduct) details from the literature. The studies that meet the minimum quality criteria and offer the details regarding these DevOps capability elements were included in this study. Further, this review included only those studies that were published in English between 2012 and 2015. This provided a sufficient coverage of recent literature on the topic in hand. The studies that did not provide the information to address the identified research question(s) were excluded from this review.

3.2. Selection of data sources and search strategies

This review included following well-known four academic and two industry electronic databases:

- IEEEXplore (www.ieeexplore.ieee.org/Xplore/)
- ACM Digital library (www.portal.acm.org/dl.cfm)
- Elsevier Science Direct (www.sciencedirect.com)
- SpringerLink (www.springerlink.com/)
- Gartner (Industry - http://www.gartner.com/)
- OVUM (Industry - http://www.ovum.com/)

The selected data sources provided sufficient literature coverage and a mix of academia and industry perspectives. The inclusion of industry databases, unlike traditional SRL studies, is a distinct feature of this review and complements the academic papers. One may argue that the industry reports tend to make more general claims, include biased views, lack a clear and sound research methodology, and peer review. Therefore, in order to avoid such uncesseary argument, industry sources are analysed separately (e.g. marked with letter 'N'). The data sources were systematically searched using the carefully selected search terms or keywords (see Table 1). For instance, we have included the term continuous delivery along with DevOps, as they were found to be very complimenting to each other. Each item from first category (i.e., "DevOps Adoption") was combined with each item from the second category (i.e., "DevOps Challenges") and each item from third and fourth category (i.e., "DevOps productivity" and "DevOps Capability" respectively). These were combined using Boolean "AND". This was done to ensure that the relevant studies are not missed.

This first stage of search and filtration resulted in a total of 3790 "hits"; however, in second stage, after filtering the duplicate papers and eliminating studies based on the titles and the keywords, the number of studies was reduced to 134. After this stage, the irrelevant studies were excluded based on the review of abstracts and the total numbers were reduced to 67. Finally, after reading all these 67 papers, we managed to found 32 papers relevant to our investigation. The papers, which were not providing relevant information about our research topic and questions in hand, were excluded. Table 2 provides the study selection stages and assessment details. RefWorks was used to manage the relevant citations from stage 1. The citations were imported into Excel sheets and the source of each study and inclusion/exclusion decisions were recorded. Table 2 provides the assessment method and criteria in detail for each stage.

Search Category	Keywords
DevOps Adoption	DevOps practices, DevOps acceptance, DevOps architecture principles, DevOps adoption framework, DevOps implementation, Continuous delivery, DevOps dimensions.
DevOps Challenges	DevOps team challenges, Challenges to DevOps success, Overcoming DevOps challenges, DevOps rewards
DevOps Productivity	DevOps quality management, DevOps processes, DevOps deployment pipeline.
DevOps Capability	DevOps capability elements, DevOps capability framework, DevOps process, DevOps roles, DevOps technology, DevOps artifacts

Table 1: Search Terms

Filtration stages	Description	Assessment criteria	Count
1st filtration (Stage 1)	Identify relevant studies from selected databases	Search category, Keywords	3790
2nd filtration (Stage 2)	Exclude duplicate studies and on the basis of titles	Title = search term(s) Yes = "accepted" No = "rejected"	134
3rd filtration (stage 3)	Exclude studies on the basis of abstracts	Abstract = provide information about selected keywords. Yes = "accepted" No = "rejected"	67
Final filtration (Stage 4)	Obtain selected and relevant papers	Address research question. Yes = "accepted" No = rejected	32

Table 2: Study Selection Method

3.3. Final quality assessment

Study screening criteria was used to ensure the relevance and quality of the selected 32 studies. The screening criteria was developed (Table 3) and applied to ensure the quality, relevance credibility, and rigor of the studies included in this research. The quality criteria questions 1-3 were used to ensure the quality of the each selected study by identifying the study's aim, objective and context. The quality criteria questions 4-5 were used to ensure the credibility of the selected studies. Finally, the last quality criteria question 6 was used to confirm the relevance of the selected study by identifying whether the study has value for research or/and the practice.

Quality criteria
1. Is the paper based on research?
2. Is there a clear statement of the aims of the research?
3. Is there an adequate description on the context in which the research was carried out
4. Was the research design appropriate for the aims of the research
5. Is there a clear statement of findings?
6. Is the study of value for research or practice?

Table 3: Quality Criteria

Table 4 provides the full details of the total number of selected studies in each phase or stage from selected databases. Most academic studies were found in the IEEEXplore and Elsevier databases comprising 53% of the total selected studies. 31% of the selected studies were obtained from the well-known industry sources such as Gartner, OVUM, and other online sources. Industry studies were included to compliment the academic studies. The selected industry papers provided valuable insights about the DevOps from the practitioners' perspectives.

In each stage of this study, three researchers searched and reported the findings to fourth researcher (experience in similar studies), who regularly reviewed the findings and provided feedback and guidance. In the end of each stage, all four researchers reviewed and discussed the agreement on the inclusion of relevant studies. All disagreements were solved by researchers' through rigorous discussions guided by the study selection method. For instance, in all hands in team meeting at the final stage, based on the review and discussion, a final set of 32 articles was obtained for detailed review.

Database/Filtration stage	Filtration stage 1	Filtration stage 2	Filtration stage 3	Final Filtration stage 4
IEEE	174	18	16	12
ACM	1065	23	7	3
Elsevier	551	25	13	5
Springer	1434	6	5	2
OVUM (Industry)	5	3	2	1
Gartner (Industry)	511	23	10	2
Others (Industry)	50	36	14	7
Count	3790	134	67	32

Table 4: Search Results

4 RESULTS

We selected and reviewed 22 relevant academic and 10 industry studies on DevOps by using the SLR approach. DevOps is relatively a new area of research and demands for the rigorous academic research. This is evident from the research results presented in this paper. For instance, most of the relevant academic studies were from different conference/workshop proceedings contributing 72% of the total 22 academic papers. Remaining 30% were from journal articles, book chapters and magazine. Most of the academic publications are workshop and conference papers, with very few journal publications. This indicates the need for more studies in this emerging area of research. Table 5 provides an overview of the selected academic studies by publication channel.

Publication Channel	Type	Study	Count
RELENG	Workshop	A1, A2	2
PESOS	Workshop	A3	1
Software, IEEE	Journal	A4	1
IPCC	Conference	A5	1
ESEM	Conference	A6	1
MILCOM	Conference	A7	1
Internet Computing	Journal	A8	1
QUDOS	Workshop	A9	1
System and Software	Journal	A10	1
ITSC	Conference	A11	1
Springer	Book Chapters	A12, A13, A22	3
RCoSE	Workshop	A14-A15	2
WICSA	Workshop	A16	1
SCC	Conference	A17	1
WOSP	Workshop	A18	1
ICPC	Conference	A19	1
Queue – Quality Assurance	Magazine	A20	1
INTECH	Conference	A21	1

Table 5: Academic Publication Channels

As discussed in section 3, the quality of each selected studies was assessed by using the quality criteria checklist (Table 3). The quality measures were adopted to ensure that a selected study will make a valuable contribution to this SLR research. The quality review results are summarised in table 6. As the selected papers came from academic and industry sources, we have used '1' for all the academic papers, and '0' for the industry papers in 'Research' (or academic) and 'Research Design' columns. All the selected studies found to have a clear statement of aims and context. All the selected academic papers clearly describe the research design to achieve the aim of the study in contrast to industry papers. All the papers clearly mention their findings and value, and hence, successfully met the criteria 5 and 6.

The overall quality score for academic studies is 6 out of 6, which is considered appropriate for this this study. Naturally, industry papers' quality score (4 out of 6) was relatively low, as they don't meet academic research papers' criteria (as expected). However, these papers were included to provide the practitioners' perspectives and ensure that important points are not overlooked.

Finally, a detailed review of the selected studies was performed to address the RQ1, RQ2 and RQ3 in the overall context of main RQ. The results of this review are reported as follows: (1) DevOps Capability Elements, (2) DevOps Challenges (3) and Possible Solutions.

	1 Research	2 Aim	3 Context	4 Resarch Design	5 Findings	6 Value	Total
A1	1	1	1	1	1	1	6
A2	1	1	1	1	1	1	6
A3	1	1	1	1	1	1	6
A4	1	1	1	1	1	1	6
A5	1	1	1	1	1	1	6
A6	1	1	1	1	1	1	6
A7	1	1	1	1	1	1	6
A8	1	1	1	1	1	1	6
A9	1	1	1	1	1	1	6
A10	1	1	1	1	1	1	6
A11	1	1	1	1	1	1	6
A12	1	1	1	1	1	1	6
A13	1	1	1	1	1	1	6
A14	1	1	1	1	1	1	6
A15	1	1	1	1	1	1	6
A16	1	1	1	1	1	1	6
A17	1	1	1	1	1	1	6
A18	1	1	1	1	1	1	6
A19	1	1	1	1	1	1	6
A20	1	1	1	1	1	1	6
A21	1	1	1	1	1	1	6
A22	1	1	1	1	1	1	6
N1	0	1	1	0	1	1	4
N2	0	1	1	0	1	1	4
N3	0	1	1	0	1	1	4
N4	0	1	1	0	1	1	4
N5	0	1	1	0	1	1	4
N6	0	1	1	0	1	1	4
N7	0	1	1	0	1	1	4
N8	0	1	1	0	1	1	4
N9	0	1	1	0	1	1	4
N10	0	1	1	0	1	1	4
Total	22	32	32	22	32	32	

Table 6: Quality Assessment

4.1 DevOps Capability Elements

RQ1: What are the key people (roles), processes, technology and artefacts (workproducts) for DevOps??

A capability can be described in terms of four key elements such as people, process, technology and artefacts (Gill 2015; ISO/ IEC 2014). These high level elements provide the theoretical lens to capture, analyse and report the DevOps capability views.

People View

People is an important element of any capability. People perform different roles, which are often referred as actors. Selected studies were carefully reviewed with the aim to identify DevOps related roles for IS delivery. Table 7 reports the identified 11 categories of roles. It can be noticed from Table 7 that a new role of "DevOps Engineer" is emerging to facilitate the integrated process of DevOps to smoothly deploy software and hardware system increments into production environment. DevOps engineers provides technical support to create standard DevOps scripts, lay out foundation for execution and guide overall DevOps teams to move in the right direction. This indicates that the effective establishment of the DevOps capability requires a new role of DevOps engineer (e.g. A7) in addition to existing traditional development and operations roles.

People (Roles)	**Papers**	**Count**
System administrator	A7,A12	2
Developer and operations	A7,A8,A9,A11	4
DevOps engineer Security engineer Infrastructure management team Tester Orchestrator Quality assurer System administrator Database administrator Network technician	A7	1

Table 7: DevOps Roles

Process View

Roles are performed by different people in DevOps process(s). A process is a set of DevOps activities that is triggered by some event. Based on the detailed review, a number of views of integrated DevOps process are identified and reported here (Table 8). These are organised into 6 categories: Full lifecycle view, continuous delivery pipeline view, continuous improvement view, multi-stage testing view, multi-stage deployment view and analytics view. Full lifecycle provides the complete coverage for the integrated DevOps process and includes activities such as plan and measure, develop and test, release and deploy, and monitor and optimise.

Continuous delivery pipeline of IS is core to DevOps and provides a partial view of the full DevOps lifecycle. It consists of build, test (e.g. acceptance test, performance test, manual test as well as code coverage) and production deployment activities (Andreas et al. 2013; Dyck et al. 2015). Any failure in the pipeline is identified and resolved as soon as possible (Fitzgerald & Stol 2015; Dlugi et al. 2015) to ensure that working software and hardware is made available to actual customers in small increments as quickly as possible (Neely & Stolt 2013; Humble & Farley 2011).

Continuous delivery pipeline involves continuous improvement, which is another view of the DevOps process identified during this literature review. Continuous improvement is enabled through continuous integration, continuous deployment, continuous testing, continuous evolution and continuous monitoring. It has been identified in this review that instead of a big bang testing and deployment, a multi-stage approach is preferred for both testing and deployment in the overall context of continuous delivery pipeline (see Table 8). Finally, the identified DevOps analytics view refers to a data-driven approach to optimising service quality, resource utilization and cost including the infrastructure optimisation.

Processes	**Description**	**Papers**	**Count**
Full Lifecycle	Plan and measure Develop and test Release and deploy Monitor and optimise	A7, A19	2
Continuous delivery pipeline	Build Acceptance test Performance test Manual test Production	A1, A2, A4, A15, A5, A16, A19, A21	8
Continuous Improvement	Continuous Integration Continuous Deployment Continuous Testing Continuous evolution Continuous Monitoring	A1, A2, A4, A5, A7, A10, A13, A21	7
Multi stage testing	Code Commit Source Control Build Deploy to Test Env. Automatic Security Test Report & Notify Publish to Release Repository Deploy to Production	A7	1
Multi-stage deployment	Deployment to test environment for final round of testing Deployment to production for small no of users Deployment to production for all users (full deployment)	A8	1
Analytics	Service quality Resource utilisation Cost	A3	1

Table 8: DevOps Processes

Technology View

DevOps processes require the support of relevant technology to enable the fast and automated delivery of working software (Azof 2011; Poppendieck & Cusumano 2012). For instance, DevOps delivery pipeline can be automated using the (1) Source Control, (2) Issue Control, (3) Build System, (4) Documentation System, (5) Code Review System, (6) Monitoring System and (7) Communication System (Gill & Bunker 2013; Cois et al. 2014). This review paper identified 23 tools that can support DevOps process automation (Table 9). Most of the reported tools are for building the systems (Build System). For instance, Git supports the version control; Chef and Puppet support the automated build and hardware provisioning process. It can be observed form this review, there are a number of different types of tools, which are different in scope. Thus, a single tool may not be sufficient for the end to end automation needs of DevOps. These tools need to be integrated with each other to create an automated integrated DevOps delivery pipeline appropriate to context in hand. Leppänen, Kilamo, & Mikkonen (2015) mention that appropriate technology is critical for enabling the effective automated DevOps. The purpose of this paper is identify the DevOps technologies, and is not to provide detailed assessment of each technology, which is beyond the scope of this paper.

Technology	**Description**	**Papers**	**Count**
Jira	Issue Control	A15	1
Trello	Idea Generation	A15	1
Jenkins	Test Automation	A1,A 2, A15	3
GIT	Version Control	A2, A15	2
CHEF	Build System	A1,A2, A15, A21	4
PUPPET	Build System	A1, A21	2
DOCKER	Build System	A2	1
Cloud Watch	Monitoring	A15	1
New Relic	Monitoring	A15	1
PingDom	Monitoring	A15	1
Zabbix	Monitoring	A15	1
PagerDuty	Feedback	A15	1
Sell	Script Language for Automation	A21	1
Perl	Script Language for Automation	A21	1
Python	Script Language for Automation	A21	1
IBM Smart Cloud	Cloud for Orchestration	A21, A22	2
AWS	Cloud for Orchestration	A1, A2, A17	3
CloudWave	Cloud for Orchestration	A3,	1
Glu	Build System	N2	1
Hudson	Build System	N2	1
Maven	Build System	N2	1
Gradle	Build System	N2	1
Ant	Build System	N2	1

Table 9: DevOps Technology

Artefact View

People apply technology-enabled automated DevOps process and generate different IS artefacts. An artefact could be a data model, executable code, hardware configuration script or integrated working software. Artefact is an important element. However, surprisingly, the identified studies do not discuss it in detail. A very few artefacts (only 5 categoories) were found, which are mentioned in Table 10. This seems to suggest that more attention is being given to DevOps people, process and technology elements in contrast to the actual deliverables or artefacts, which is contrary to the "working system or artefact" agile principle.

Artefacts	**Description**	**Papers**	**Count**
Log Files	A way to deliver metrics about the application in production. Stored in shared file system. Supports feedback driven mechanism.	A13	1
Shippable artefacts	These types of artefacts are continuously produced and shipped into productions by the DevOps pipeline. Each artefact has a unique identifier and version to enable traceability.	A1	1
Business environment change artefacts	These artifacts form a part of continuous planning where any strategic change to business environment is recorded and reported via such artifacts.	A10	1
Artefacts in continuous integration	These artefacts capture information about the continuous integration details with their success or failure status. .	A10	1
Infrastructure artefacts	These artefacts are core to the integrated DevOps and comprise of scripts for automatically provisioning of hardware infrastructure (e.g. infrastructure as a code) for software development, testing and release.	A13	1

Table 10: DevOps Artefacts

4.2 DevOps Challenges

RQ2: What are the key challenges in adopting DevOps?

The adoption of DevOps is not straight forward and may pose several challenges. Thus, this section, building on the previous section, reviewed and identified DevOps challenges (RQ2). This review identified seven key challenges (Table 11), which are organised into people and culture, security, misunderstanding, process and technology challenges. Please note that people, culture and lack of understanding of DevOps were found to be the important challenges compared to others (based on count value). However, count is one indicator and may not necessarily mean it is actually the most important challenge in DevOps. For instance, resistance to change due to change or loss of job fear is a long standing people and culture issue (e.g. C1). Siloed lines of business found to be challenging where business units do not interact with each other until absolutely necessary. This separation was typically created because of the lack of desire to work outside the drawn lines. Due to which there have been naturally formed certain groups of people that may not like to collaborate to each other yet are not working towards the same goal (e.g. C1, C2). Sense of responsibility is another people and culture challenge, which highlights that developers were not found to show accountability for the deployment of the software or a product which makes further harder to adopt DevOps.

Security challenge draws our attention to the key question (subject to further research): how to ensure the security of the automated software and hardware DevOps pipeline? Misunderstanding is another important challenge. This highlights that the biggest mistake IT organisations commit, while adopting DevOps, is to try to solve the problem with a prescriptive methodology or a big expensive all-in-one DevOps technology. Finally, process and technology challenge highlights another area of research: how to re-engineer the schism that exists between development and operations processes. This clearly indicates that DevOps is not a just technology or service that can be downloaded (from an open source or vendor) and can be used to fasten the software development life cycle process. There is a need to clearly understand and address the DevOps as a whole for its effective and less risky adoption.

SN	Challenge Category	Challenge	Papers	Count
C1	People and Culture	Resistance to change	A4, A5, A11, A12, A16, N3, N6, N8	8
C2	People and Culture	Siloed lines of business	A3, A4, A5, A11, A12, A13, N1, N3, N6, N8	10
C3	People and Culture	Sense of responsibility.	A4, A7, A11, A12, A13, N5, N6, N8	8
C4	Security	Automated DevOps pipeline security	A2, A11, N2, N4	4
C5	Misunderstanding	All in one tool	A10, N3, N6, N8	4
C6	Misunderstanding	Moving faster means compromising the quality	A4, A10, A11, N5, N6, N8	6
C7	Process and Technology	Re-engineer the schism between Development and Operations	A7, A10, N6, N8	4

Table 11: DevOps Challenges

4.3 Solutions

<u>**RQ3:** What are the possible solutions to the challenges in adopting DevOps?</u>

This study identified a set of possible solutions (6 solution categories) to address the identified DevOps adoption challenges, which are mapped in Table 12. The subset of literature that was analyzed in our study suggests that the adoption of DevOps should be planned and executed as a transformational change to effectively developing a collaborative environment and adoption roadmap to deal with the inherent challenges such as resistance to change and siloed lines of business (see C1-C3, C7). Determining a gap between current and required collaborative culture state could help building a roadmap and implementation plan for smoother and gradual cultural change for the integration of siloed Dev and Ops (e.g. A7, A10). Ongoing awareness, training and coaching (along with the effective recruitment practices) could smoother the overall DevOps adoption (e.g. A8, A9). The integrated governance could help in addressing the challenge of DevOps pipeline security monitoring and siloed business units by having clear integrated roles, co-ownership, responsibility and accountability (e.g. A13). Harvesting cross-functional processes and using appropriate technology could result in an overall collaborative automated DevOps environment (e.g. A7, A10).

Challenges	Solution	Papers	Count
C1, C2, C7	Recognizing DevOps as a transformative change	A7, A10, N7, N9, N10	5
C2, C3	Developing collaborative environment and adoption roadmap	A7, A10, N5, N10	4
C1-C7	Introducing better training and coaching	A7, A8, A10, N9, N10	5
C1-C7	Enabling continuous improvement	A7, A10, N5, N7, N9	5
C3, C4	Establishing effective DevOps governance	A7, A10, A13, N7, N9, N10	6
C7	Harvesting technology-enabled cross-functional collaborative environment	A7, A8, A10, N7, N9, N10	6

Table 12: DevOps Challenges and Potential Solutions

5 DISCUSSION

This paper attempted to provide a state-of-the-art and knowledge-base covering key DevOps capability roles, processes, technologies, artefacts, challenges and solutions by using a well-known SRL method. This review reported a number of roles, in particular the emerging role of the DevOps engineer (Table 7), in the overall context of people and organization. DevOps engineer role is like a glue between Dev and Ops. It provides necessary technical support to DevOps teams. DevOps teams should focus on delivery working software rather juggling with technology hardware issues (e.g. A7). Thus, it is important to note here that development teams take ownership to the larger extent to develop and deploy software increments by themselves in production. The review results also draw our attention to the need of fostering the culture of collaboration and communication for bridging the gap between the development and technology operations teams (Gottesheim 2015). This research identified two key aspects of DevOps process. These aspects focus on continuous delivery pipeline and continuous improvement (see Table 8). It is not all about quickly pushing solutions through the delivery pipeline, rather there is a sense of quality through the use of feedback loops and DevOps analytics. In short, it has been identified that the continuous delivery pipeline and continuous improvement are the most important aspects (see Section 4) of the overall integrated DevOps lifecycle. Organizations should pay close attention to these process aspects when considering the adoption of DevOps processes within the overall context of DevOps capability.

Traditional documentation driven development and operations could be slow and problematic. DevOps processes (Table 8) focuses on continuous and frequent development, testing and deployment of working software in production. The automation of DevOps processes will ensure that the important steps are not overlooked due to human error, and teams have the ability to orchestrate and track the DevOps delivery pipeline. The use of technology does not only enable (Virmani 2015) the rapid continuous delivery but it also provides an opportunity to channel continuous feedback from end-users to developers for continuous improvement. Such fast user feedback cycles enable producing quality products. The use of monitoring tools enables capturing and inspecting issues, and resolves any poor software quality related user concerns as soon as possible (Waller et al. 2015).

The use of technology also allows to quickly develop proof of concepts for complex systems and test new ideas to help business in making more informed decisions based on actual end-user feedback and behaviour. Thus, use of technology and frequent feedback cycles precisely enhance user engagement and inform about the actual needs of the users (Leppänen, Kilamo, & Mikkonen 2015). Although, agile approaches value people over processes and tools, however, based on the review results, it can be suggested that technology is not a support element or low value element (Tables 7-9), rather it is core to the contemporary people-centric DevOps. Deliverables or artefacts (Table 10) are important to determine the productivity of a DevOps pipeline. However, surprisingly, the identified studies do not discuss it in much detail. This indicates the need of more research in this area.

The bulk of this review is focused on the core DevOps capability elements, which provide a knowledge-base to guide the informed adoption of DevOps for IS. DevOps is not all about technology and this is evident from the identified challenges (Table 11). Highly discussed challenges are around People, culture and lack of understanding of DevOps (see Section 4). This suggests that DevOps adoption approach should be people-centric as opposed to merely a new technology adoption initiative. It has been observed that most of the suggested solutions advise assembling cross-functional teams with different functional expertise including defining clear roles and providing appropriate training (Table 12). The key highlight is that adopting DevOps is not an ad-hoc or routine operational change, it is transformative in nature and requires fundamental shift in the traditional ways of working (e.g. C1, C3, C7, and A7). It requires developing the supporting culture and have the right people appropriate to organizational context (Farroha & Farroha 2014). It is not one-off adoption or big bang change, rather, the focus should be on designing, implementing and continually improving the DevOps roadmap (e.g. C3, A10).

In a nutshell, this SLR confirms the importance of studying DevOps, especially its practical adoption. Academia and industry may use these study findings as a foundation for developing theoretical and practical situation-specific DevOps frameworks for desgining the IS delivery pipeline. For instance, researchers may be interested to empirically study the impact of DevOps on IS project performance, which is deemed as a theoretical gap. The identified challenges and solutions can be used by commercial organizations to develop a less risky and informed approaches for DevOps. Technology vendors may be interested to further understand the integrated elements of DevOps capability (e.g. people, process, tools and artefacts) and develop appropriate end-to-end technology platforms for the holistic adoption. Security professionals and researchers may be interested in developing new techniques for DevOps security. Further, advanced analytics professionals and researchers may be interested to develop new algorithms and solutions for integrated DevOps analytics for enabling continuous adaptation.

This research clearly has implications for both research and practice. However, like many other studies, it has its boundary and limitations. Given the scope and time constraints of the research project, we only used selected well-known databases both from academia and industry. However, we made sure that the selected databases provide us sufficient recent quality literature to address the research topic in hand. To make sure that the selection process remains unbiased, we identified keywords and search terms directly from the research questions. Furthermore we have utilized the multistage process to document the reasons for inclusion and exclusion at each stage. Apparently, like any other SLR study, it is impossible to claim that the selected keywords and search strings have not caused any omission of relevant studies. Every attempt was made to ensure the coverage of the literature, therefor, in addition to academic databases, we also used two well-known industry databases. This provided us a good mix of academic and industry articles, which are explicity highlighted in the analsyis. The analysis and projection of concepts and categories are subject to human errors, which may lead to inconsistencies. In order to deal with the researcher bias and any human error reduction, regular meetings and internal peer review feedback among the researchers were conducted. This was done to ensure the quality of the research process and findings.

6 CONCLUSION

DevOps is an emerging topic of greater interest among IS community. Interest is there, however, the challenge is how to effectively establish an integrated DevOps capability for effectively delivering IS including software and hardware that support data-intensive applications. This requires the clear understanding of underlying elements and challenges (including potential solutions) of DevOps capability. To to do so, this paper presented a systematic review of DevOps literature published in selected academic (22 papers) and industry (10 papers) databases using a SLR approach. We found 11 major roles, 6 processes, 23 technolgies, 5 artefacts, and 7 challenges (including 6 corresponding solutions). DevOps engineer is being the newly identified role. Continuous delivery pipeline and continuous improvement are most highlighted major DevOps processes. Build system technology is being the key focus of the DevOps. Finally, major challenges are around people and culture, and the misunderstanding of the DevOps. Some potential research gaps such as minimal literature on DevOps artifacts, DevOps analytics and DevOps tool-chain integration, which could be future study areas. The review results will serve as a knowledge-base, which can be cast into developing theoretical and practical frameworks for DevOps adoption for a particular context.

APPENDIX A. SELECTED STUDIES (ACADEMIC)

A1	V. Armenise, "Continuous delivery with jenkins: Jenkins solutions to implement continuous delivery," in *Release Engineering (RELENG), 2015 IEEE/ACM 3rd International Workshop on,* 2015, pp. 24-27.
A2	L. Bass, R. Holz, P. Rimba, A. B. Tran and L. Zhu, "Securing a deployment pipeline," in *Release Engineering (RELENG), 2015 IEEE/ACM 3rd International Workshop on,* 2015, pp. 4-7.
A3	D. Bruneo, F. Longo, G. Merlino, N. Peditto, C. Romeo, F. Verboso and A. Puliafito, "Enabling collaborative development in an OpenStack testbed: The CloudWave use case," in *Proceedings of the Seventh International Workshop on Principles of Engineering Service-Oriented and Cloud Systems,* 2015, pp. 24-30.
A4	L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too," *Software, IEEE,* vol. 32, pp. 50-54, 2015.
A5	C. A. Cois, J. Yankel and A. Connell, "Modern DevOps: Optimizing software development through effective system interactions," in *Professional Communication Conference (IPCC), 2014 IEEE International,* 2014, pp. 1-7.
A6	F. Erich, C. Amrit and M. Daneva, "Cooperation between information system development and operations," in *8th International Symposium on Empirical Software Engineering and Measurement,* pp. 1.
A7	B. Farroha and D. Farroha, "A framework for managing mission needs, compliance, and trust in the DevOps environment," in *Military Communications Conference (MILCOM), 2014 IEEE,* 2014, pp. 288-293.
A8	D. G. Feitelson, E. Frachtenberg and K. L. Beck, "Development and deployment at Facebook," *IEEE Internet Comput.*, pp. 8-17, 2013.
A9	N. Ferry, F. Chauvel, H. Song and A. Solberg, "Continous deployment of multi-cloud systems," in *Proceedings of the 1st International Workshop on Quality-Aware DevOps,* 2015, pp. 27-28.
A10	B. Fitzgerald and K. Stol, "Continuous software engineering: A roadmap and agenda," *J. Syst. Software,* 2015.
A11	S. W. Hussaini, "Strengthening harmonization of development (dev) and operations (ops) silos in IT environment through systems approach," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on,* 2014, pp. 178-183.
A12	M. Hüttermann, "Beginning DevOps for developers," in *DevOps for Developers*Anonymous Springer, 2012, pp. 3-13.
A13	M. Hüttermann, "Gain fast feedback," in *DevOps for Developers* Anonymous Springer, 2012, pp. 81-94.
A14	S. Krusche, L. Alperowitz, B. Bruegge and M. O. Wagner, "Rugby: An agile process model based on continuous delivery," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering,* 2014, pp. 42-50.
A15	M. Leppänen, T. Kilamo and T. Mikkonen, "Towards post-agile development practices through productized development infrastructure," in *Proceedings of the Second International Workshop on Rapid Continuous Software Engineering,* 2015, pp. 34-40.
A16	Lianping Chen, "Towards architecting for continuous delivery," in *Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on,* 2015, pp. 131-134.
A17	M. A. McCarthy, L. M. Herger, S. M. Khan and B. M. Belgodere, "Composable DevOps: Automated ontology based DevOps maturity analysis," in *Services Computing (SCC), 2015 IEEE International Conference on,* 2015, pp. 600-607.
A18	J. F. Perez, W. Wang and G. Casale, "Towards a DevOps approach for software quality engineering," in *Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development,* 2015, pp. 5-10.
A19	N. Rathod and A. Surve, "Test orchestration a framework for continuous integration and continuous deployment," in *Pervasive Computing*

	(ICPC), 2015 International Conference on, 2015, pp. 1-5.
A20	J. Roche, "Adopting DevOps practices in quality assurance," *Commun ACM,* vol. 56, pp. 38-43, 2013.
A21	M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," in *Innovative Computing Technology (INTECH), 2015 Fifth International Conference on,* 2015, pp. 78-82.
A22	J. Wettinger, U. Breitenbücher and F. Leymann, "DevOpSlang–bridging the gap between development and operations," in *Service-Oriented and Cloud Computing*Anonymous Springer, 2014, pp. 108-122.

APPENDIX B. SELECTED STUDIES (NON-ACADEMIC)

N1	Alex Honor, 2014, " Common Objections to DevOps from Enterprise Operations", Dev2OPS, viewed 19 October 2015, <http://dev2ops.org/2014/06/adopting-devops-in-enterprise-operations/>
N2	Azof, M. 2011, "DevOps advances in Release Management and Automation", OVUM, viewed 20 October 2015, <http://www.ovum.com/research/devops-advances-in-release-management-and-automation/>
N3	Carey, L. 2014, "The biggest challenges facing DevOps", Jaxenter, viewed 22 October 2015, <https://jaxenter.com/the-biggest-challenges-facing-devops-107634.html>
N4	CBR 2015, "Data security poses big threat to DevOps adoption", CBRonline, viewed 20 October 2015, <http://www.cbronline.com/news/big-data/software/data-security-posing-threat-to-devops-adoption-210815-4652396 >
N5	Drew, S. 2015, "New Gartner Report Highlights the Trend Toward DevOps", VeraCode, viewed 20 October 2015, <https://www.veracode.com/blog/2015/06/new-gartner-report-highlights-trend-toward-devops-sw >
N6	Hand, J. 2015, "The IT Culture War: The Struggle to Adopt DevOps", Wired, viewed 22 October 2015, <http://www.wired.com/insights/2015/03/culture-war-struggle-adopt-devops/ >
N7	Horvath, K. 2015, "DevOps Part 2: Methods, Practices and Tools", IntLand, viewed 21 October 2015, <http://intland.com/blog/agile/devops/devops-part-2-methods-practices-and-tools/ >
N8	Riley, C. 2014, "6 Challenges Facing DevOps & Operations Teams in 2015", Logentries, viewed 19 October 2015, <https://blog.logentries.com/2014/10/6-challenges-facing-devops-and-operations-teams-in-2015 />
N9	Spafford, G., Head, I., Bandopadhyay, I. 2015, "Survey Analysis: DevOps Adoption Survey Results", Gartner, viewed 20 October 2015, <http://www.gartner.com/document/3139918?ref=solrAll&refval=157498997&qid=eb9511e33e8aa62e4c3b1a668e7c0d50 >
N10	This is what good looks like, 2014, "Overcoming challenges facing adoption of a DevOps mindset", Thisiswhatgoodlookslike, viewed 22 October 2015, <http://thisiswhatgoodlookslike.com/tag/overcoming-challenges-facing-adoption-of-a-devops-mindset/ >

REFERENCES

1. Agile Manifesto.2001. Manifesto for Agile Software Development, .http://www.agilemanifesto.org/. 2001, Accessed 14 March 2015.
2. Bayser, M., Azevendo, L., Cerquiera, R. 2015, "ResearchOps: The Case for DevOps in Scientific Applications", 10th International Workshop on Business-driven IT Management, IFIP/IEEE, pp. 1398-1404
3. Bellomo, S. 2014, "Architectural Implications of DevOps." CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
4. Bosch, J. 2012, "Building products as innovation experiment systems" in *Software Business* Springer, pp. 27-39.
5. Claps, G.G., Svensson, R.B. & Aurum, A. 2015, "On the journey to continuous deployment: Technical and social challenges along the way", *Information and Software Technology,* vol. 57, pp. 21-31.
6. Dybå, T. & Dingsøyr T. 2008, "Empirical studies of agile software development: A systematic review", Inf. Soft. Techno., vol. 50, pp. 833-859.
7. Fagerholm, F., Guinea, A.S., Mäenpää, H. & Münch, J. 2014, "Building blocks for continuous experimentation", *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*ACM, , pp. 26.
8. Gill, A.Q. 2015, "Agile enterprise architecture modelling: Evaluating the applicability and integration of six modelling standards", Information and Software Technology, vol. 67, pp. 196-206.
9. Gottesheim, W. 2015, "Challenges, Benefits and Best Practices of Performance Focused DevOps", Proceedings of the 4th International Workshop on Large-Scale Testing ACM, , pp. 3.
10. Httermann, M. 2012, *DevOps for developers,* Apress, Berkely, USA
11. Humble, J. and Farley, D. 2011, "*Continuous delivery"*, Upper Saddle River, NJ: Addison-Wesley.
12. ISO/IEC 2014. Software Engineering - Metamodel for Development Methodologies (SEMDM). ISO/IEC Standard 24744.: 2014, ISO, Geneva
13. Kitchenham, B. & Charter, S. 2007,"Guidelines for performing Systematic Literature Reviews in Software Engineering", EBSE Technical Report, ver. 2.3
14. Mueller, E., Wickett, J., Gaekwad, K., Karayanev, P., What is DevOps?, http://theagileadmin.com/what-is-DevOps/, 26.10.2014.
15. Y. Liu, C. Li, W. Liu, Integrated Solution for Timely Delivery of Customer Change Request: A Case Study of Using DevOps Approach, International Journal of U- and E-Service Science and Technology, vol. 7, No. 2, pp. 41-50, 2014.
16. Neely, S. & Stolt, S. 2013, "Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy)", *Agile Conference (AGILE), 2013* IEEE, , pp. 121.
17. Patwardhan, B. 2012, "An Overview of DevOps", *CSI Communications,* , pp. 14.
18. Peuraniemi, T., 2014, 'Review: DevOps, value-driven principles, methodologies and tools, Data- and Value-Driven Software Engineering with Deep Customer Insight', PROCEEDINGS OF THE SEMINAR NO. 58314308 pp 43-49.
19. Poppendieck, M. & Cusumano, M. 2012, "Lean software development: A tutorial", *Software, IEEE,* vol. 29, no. 5, pp. 26-32.
20. Gill, A.Q. & Bunker, D. 2013, "Towards the development of a cloud-based communication technologies assessment tool: An analysis of practitioners' perspectives", *VINE,* vol. 43, no. 1, pp. 57-77.
21. Rackspace 2014, 'DEVOPS ADOPTION, 2014
22. Rowe, F. 2014, " What literature review is not: diversity, boundaries and recommendations. *European Journal of Information Systems"*, pp.241-255.
23. Williams, L. 2012, "What agile teams think of agile principles", *Communications of the ACM,* vol. 55, no. 4, pp. 71-76.
24. Zhong, B., Ding, L., Luo, H., Zhou, Y., Hu, Y. and Hu, H. 2012, "Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction"*, pp.58-70.
25. Hüttermann, M. 2012, "Building Blocks of DevOps." *DevOps for Developers*. Apress, pp. 33-47.
26. Waller, J., Nils C, E. and Wilhelm H. 2015, "Including performance benchmarks into continuous integration to enable DevOps." *ACM SIGSOFT Software Engineering Notes*, vol. 40.2, pp. 1-4.
27. Andreas S., Marc, R., and Dietmar, F. 2013, "Continuous integration and automation for DevOps.", *IAENG Transactions on Engineering Technologies*, Springer Netherlands, pp. 345-358.
28. Dyck, A., Ralf, P. and Horst, L. 2015, "Towards definitions for release engineering and DevOps", *Proceedings of the Third International Workshop on Release Engineering*, IEEE Press
29. Wettinger, J., Vasilios, A. and Frank L. "Automated Capturing and Systematic Usage of DevOps Knowledge." *Proceedings of the IEEE International Conference on*. IEEE Computer Society.
30. Stillwell, M. and Jose, C. 2015, "A DevOps approach to integration of software components in an EU research project." *Proceedings of the 1st International Workshop on Quality-Aware DevOps*. ACM.
31. Dlugi, M., Andreas, B. and Helmut, K. 2015, "Model-based performance evaluations in continuous delivery pipelines." *Proceedings of the 1st International Workshop on Quality-Aware DevOps*. ACM.
32. Erich, F., Chintan, A. and Maya, D. 2014,"Report: DevOps Literature Review."
33. Bass, L., Weber, I. and Zhu, L. DevOps: A Software Architect's Perspective. Addison-Wesley, 2015.